



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV RADIOELEKTRONIKY

DEPARTMENT OF RADIO ELECTRONICS

PROGRAM PRO POROVNÁVÁNÍ KŘIVEK METODOU IELF

COMPARATIVE TOOL FOR DATASETS USING METHOD IELF

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Zdeněk Podjukl

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Petr Kadlec, Ph.D.

BRNO 2018

Bakalářská práce

bakalářský studijní obor **Elektronika a sdělovací technika**

Ústav radioelektroniky

Student: Zdeněk Podjukl

ID: 186165

Ročník: 3

Akademický rok: 2017/18

NÁZEV TÉMATU:

Program pro porovnávání křivek metodou IELF

POKYNY PRO VYPRACOVÁNÍ:

Seznamte se s metodou IELF (Integrated Error against Log Frequency). Tato metoda umožňuje porovnávání datových křivek, např. naměřených a simulovaných hodnot. Metodu IELF implementujte ve formě funkce v Matlabu. Vytvořte set křivek pro ověření s různou mírou podobnosti.

V Matlabu vytvořte GUI (Graphical User Interface), které bude umožňovat načítat data pro porovnání ve formě různých datových formátů (*.txt, *.dat, *.xls, *.mat a další). Nástroj bude následně vizualizovat výsledky porovnání metodou IELF. Vytvořte dokumentaci k vzniklému programu.

DOPORUČENÁ LITERATURA:

[1] SIMPSON, R.J., C.R. JONES, I. MACDIARMID, A. DUFFY a D. COLEBY. The integrated error against log frequency (IELF) method for CEM validation. In: 2005 International Symposium on Electromagnetic Compatibility, 2005. EMC 2005 [online]. IEEE, 2005, s. 296-300 [cit. 2017-05-11]. DOI: 10.1109/ISEMC.2005.1513517. ISBN 0-7803-9380-5. Dostupné z: <http://ieeexplore.ieee.org/document/1513517/>

[2] SMITH, Scott T. Matlab advanced gui development. Indianapolis: Dog Ear, 2006. ISBN 1-59858-181-3.

Termín zadání: 5.2.2018

Termín odevzdání: 24.5.2018

Vedoucí práce: Ing. Petr Kadlec, Ph.D.

Konzultant:

prof. Ing. Tomáš Kratochvíl, Ph.D.
předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

ABSTRAKT

Tato práce se zabývá realizací programu v prostředí MATLAB pro výpočet míry shody dvou křivek metodou IELF (Integrated Error Against Logarithmic Frequency). Tato metoda se používá pro porovnávání dvou křivek. Většinou se s ní setkáváme v souvislosti s EMC (Electromagnetic Compatibility) problematikou. Práce je rozdělena na 4 kapitoly. První se zabývá porovnávacími metodami, druhá podobou výsledného grafického rozhraní, třetí popisuje program obsluhující celou aplikaci a poslední je věnována testování správnosti výpočtů.

KLÍČOVÁ SLOVA

Porovnávání, IELF, EMC, opakovatelnost

ABSTRACT

This thesis deals with a realization of MATLAB IELF method solver. IELF method was designed for comparison of two datasets. The method is mostly used in EMC (Electromagnetic Compatibility) measurements and simulations. Thesis consists of 4 chapters. First one showcases some of the most widely used comparison methods, the second describes looks and behavior of the application, third contains description of program code and the last one deals with calculations testing.

KEYWORDS

Comparison, IELF, EMC, repeatability

PODJUKL, ZDENĚK. Program pro porovnávání křivek metodou IELF. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav radioelektroniky, 2018. 25 s. Bakalářská práce. Vedoucí práce: Ing. Petr Kadlec, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svoji semestrální práci na téma Program pro porovnávání křivek metodou IELF jsem vypracoval samostatně pod vedením vedoucího semestrální práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené semestrální práce dále prohlašuji, že v souvislosti s vytvořením této semestrální práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne

.....

(podpis autora)

PODĚKOVÁNÍ

Tímto bych rád poděkoval svému vedoucímu práce Ing. Petrovi Kadlecovi Ph.D., který mi věnoval spoustu svého času a pomáhal mi jak s formálními náležitostmi práce, tak i s řešením problémů při psaní kódu.

OBSAH

Seznam obrázků	vi
Úvod	1
1 Porovnávací Metody	2
1.1 Metoda IELF	2
1.2 Korelace	3
1.3 Feature Selective Validation (FSV)	4
2 Grafické uživatelské rozhraní	5
2.1 Podoba grafického rozhraní	6
2.2 Vstupní soubory	6
2.3 Frekvenční pásma	8
2.4 Spuštění výpočtu a výpis výsledků	8
3 Program v pozadí	9
3.1 Gui.m	10
3.1.1 Callbacky <i>sb1_call</i> a <i>sb2_call</i>	10
3.1.2 Callbacky <i>file_1_call</i> a <i>file_2_call</i>	11
3.1.3 Callback <i>add_button_call</i>	11
3.1.4 Callback <i>delete_button</i>	11
3.1.5 Callbacky <i>clearall_call</i> a <i>clearlog_call</i>	11
3.1.6 Callback <i>reset_call</i>	11
3.1.7 Callback <i>scaling</i>	11
3.1.8 Callback <i>go_call</i>	12
3.2 Funkce <i>inputPlot.m</i>	13
3.3 Funkce <i>dataRead.m</i>	14
3.4 Funkce <i>interpolation.m</i>	15
3.5 Funkce <i>addToLog.m</i>	16
3.6 Funkce <i>bandCheck.m</i>	17
3.7 Funkce <i>bandLimits.m</i>	18
3.8 Funkce <i>ieIfCalc.m</i> a <i>ieIfModCalc.m</i>	18
3.9 Funkce <i>mainFunc.m</i>	19

4	Ověření správnosti	21
4.1	Popis vzorového souboru.....	21
4.2	Testování.....	22
5	Závěr	23
	Literatura	24
	Seznam zkratek	25

SEZNAM OBRÁZKŮ

Obrázek 1	Výsledná podoba aplikace	6
Obrázek 2	Zadávání vstupních dat	7
Obrázek 3	Obsah používaných struktur	9
Obrázek 3	Blokové schéma funkce <i>Gui.m</i>	10
Obrázek 5	Blokové schéma funkce <i>inputPlot.m</i>	14
Obrázek 6	Blokové schéma funkce <i>interpolation.m</i>	16
Obrázek 7	Blokové schéma funkce <i>mainFunc.m</i>	19
Obrázek 8	Náhled souboru pro porovnání	21

ÚVOD

Vizuální porovnávání dat s vysokou hustotou hodnot na ose x (typicky frekvenční), s čímž se setkáváme například při simulaci a měření dat pro elektromagnetickou kompatibilitu, je velmi složité z důvodu nepřehlednosti takového průběhu. Posouzení míry shody měřených a simulovaných dat (nebo obecně dvou průběhů) vyžaduje v těchto podmínkách značné zkušenosti a hluboké znalosti problematiky i konkrétního zařízení. Pro tyto účely byla vyvinuta metoda IELF. Umožňuje opakovatelné a spolehlivé porovnávání dvou průběhů a vylučuje nezanedbatelný subjektivní lidský faktor. Výhodou je také možnost snadného sdílení a srovnávání získaných dat mezi partnerskými organizacemi, což by při posuzování dat lidmi bylo neprůkazné, protože každý odborník hodnotí získané informace jinak.

Cílem této práce je tedy vytvořit program pro výpočet míry shody dvou křivek pomocí metody IELF v rozšířeném prostředí MATLAB.

Tato práce je dělena na čtyři hlavní části. První část obsahuje teoretický popis některých prakticky používaných metod pro porovnání křivek či výsledků měření. Nachází se zde i popis metody IELF a vzorce podle kterých jsou koeficienty počítány. Druhá část popisuje podobu aplikace a možnosti a limitace použití z hlediska uživatele. Třetí se zabývá programem, který obsluhuje celou aplikaci a poslední čtvrtá kapitola obsahuje popis kontroly správnosti prováděných výpočtů.

1 POROVNÁVACÍ METODY

Tato kapitola obsahuje popis několika metod prakticky používaných pro porovnávání datových křivek, výsledků měření se simulovanými daty apod.

1.1 Metoda IELF

Metoda IELF (z angličtiny The Integrated Error against Logarithmic Frequency) je metoda navržená pro porovnávání křivek s vysokou hustotou hodnot (obvykle na frekvenční ose), což není prakticky možné provést vizuálním porovnáním. Mezi hlavní přednosti této metody patří:

- spolehlivé a opakovatelné porovnávání dvou průběhů s vyloučením lidského chybového faktoru
- snadné porovnávání výsledků získaných například z několika prototypů ve snaze optimalizovat zařízení a přiblížit reálné hodnoty a vlastnosti hodnotám simulovaným (požadovaným)
- možnost snadného sdílení získaných dat mezi spolupracujícími organizacemi, neboť každý odborník hodnotí na rozdíl od jasně daného algoritmu získaná data jinak.

Metoda nepoužívá žádné derivace a je proto vhodná pro srovnávání dat obsahujících rychle se měnící hodnoty, které se můžou na první pohled nezkušenému pozorovateli jevit jako šum. Takovéto průběhy jsou v oblasti EMC naprosto běžné.

Metoda je založena na předpokladu, že při porovnávání dat s vysokou hustotou hodnot je nejdůležitějším faktorem k posouzení funkce rozdílu hodnot dvou porovnávaných křivek. Sečtením všech rozdílů hodnot v daném frekvenčním rozsahu (přičemž osa frekvence je logaritmická) získáme jediné číslo reprezentující míru shody zadaných křivek. Metodu je také možné použít pro výpočet více nezávislých frekvenčních pásem, místo počítání pro celý frekvenční rozsah. Základní rovnice používaná pro výpočet metody IELF je následující [1]:

$$IELF = \frac{\sum_{i=1}^{n-1} |error_i| \cdot (\ln(f_{i+1}) - \ln(f_{i-1}))}{\ln(f_n) - \ln(f_0)} \quad (1)$$

kde f_i reprezentuje body na frekvenční ose porovnávaných křivek, přičemž i nabývá hodnot od 1 do $n-1$ a $|error_i|$ je rozdíl hodnot porovnávaných křivek v bodě i .

V praxi se používá také lehce modifikovaná metoda, která spočívá ve sčítání bodů ležících uprostřed mezi naměřenými hodnotami ve snaze zlepšit aproximaci. Koeficient modifikované metody IELF se určuje podle rovnice [1]:

$$IELF_{mod} = \frac{\sum_{i=1}^{n-1} |error_i| \cdot \left(\ln\left(\frac{f_{i+1} + f_i}{2}\right) - \ln\left(\frac{f_i + f_{i-1}}{2}\right) \right)}{\ln(f_n) - \ln(f_0)} \quad (2)$$

V obou případech platí, že vyjde-li výsledná hodnota nula, jsou porovnávané křivky naprosto shodné. Hodnota výsledku není shora nijak omezena, tzn. teoreticky výsledek nabývá hodnot 0 až ∞ , prakticky však v rozmezí asi 0 až 18.

Metoda IELF byla testována ve spolupráci s vybranými odborníky z oblasti EMC. Odborníkům byly předloženy dvojice křivek a byli požádáni o vlastní posouzení míry shody reprezentované jedinou cifrou od 0 do 10. Jejich odpovědi se téměř ve všech případech shodovaly s výsledky získanými výpočtem podle metody IELF [1]. Výsledky metody IELF vyšly vždy trochu horší než odhad odborníků, tato odchylka byla ale ve všech případech podobná, jedná se v podstatě o soustavnou chybu, což není podstatné, protože při porovnávání jednotlivých měření bude metoda IELF použita vždy stejně a chyba tak bude eliminována. Proběhlo také testování metody IELF proti ostatním metodám užívaným k porovnávání průběhů. I zde se metoda IELF lépe shodovala s názorem panelu odborníků a to při porovnávání jak průběhů s vysokou hustotou dat tak i s ostatními průběhy.

1.2 Korelace

Korelace vyjadřuje vzájemný vztah mezi dvěma průběhy nebo procesy a používá se pro zjištění, zda je jedna zkoumaná veličina lineárně závislá na druhé.

Vzorec pro výpočet Pearsonova korelačního koeficientu je [2]:

$$R = \frac{\sum xy - \frac{(\sum x \sum y)}{n}}{\sqrt{\left(\sum x^2 - \frac{(\sum x)^2}{n}\right) \left(\sum y^2 - \frac{(\sum y)^2}{n}\right)}} \quad (3)$$

kde x a y jsou porovnávané průběhy a n je celkový počet bodů v obou průbězích dohromady. Výpočet je podmíněn znalostí konkrétního rozdělení pravděpodobnosti, což se v praxi příliš často nestává a je to jedna z hlavních nevýhod Pearsonova korelačního koeficientu. Pearsonův korelační koeficient nabývá hodnot od -1 do +1, pokud nastane případ $R = 1$, pak jsou porovnávané křivky shodné, naopak výsledek -1 značí nepřímou závislost, tzv. antikorelaci. Vyjde-li 0, jedná se o nekorelovanost, mezi zkoumanými průběhy tedy není žádná lineární závislost.

Někdy se také používá Spearmanův korelační koeficient, který je definován takto [2]:

$$R = 1 - \frac{6 \times \sum(D^2)}{n(n^2 - 1)} \quad (4)$$

Přičemž D je rozdíl v pořadí hodnot x a y a n je stejně jako v předchozím případě celkový počet bodů v obou průbězích. Spearmanův korelační koeficient se od Pearsonova liší v tom, že popisuje, jak dobře vzájemný vztah dvou průběhů odpovídá monotónní funkci.

Hlavní výhodou obou více zmíněných korelací je fakt, že jsou velmi rozšířené a známé. Společnou nevýhodou je především to, že výsledné hodnoty nenesou příliš mnoho

informace a metoda je nepřesná pro křivky s vysokou hustotou hodnot.

1.3 Feature Selective Validation (FSV)

Tato metoda byla vytvořena tak, aby lépe fungovala v případech, kde korelace zaostává. Spočívá v provedení Fourierovy transformace a následné zpětné Fourierově transformace pro nejnižších 25% dat a nejvyšších 75% zvlášť. Toto se pak dosazuje do následujících vztahů [1]:

$$ADM(f) = \left| \frac{|Lo_1(f)| - |Lo_2(f)|}{\frac{1}{N} \sum_{i=1}^N (|Lo_1(i)| + |Lo_2(i)|)} \right| \quad (5)$$

$$FDM_1(f) = \left| \frac{|Lo_1'(f)| - |Lo_2'(f)|}{\frac{2}{N} \sum_{i=1}^N (|Lo_1'(i)| + |Lo_2'(i)|)} \right| \quad (6)$$

$$FDM_2(f) = \left| \frac{|Hi_1'(f)| - |Hi_2'(f)|}{\frac{6}{N} \sum_{i=1}^N (|Hi_1'(i)| + |Hi_2'(i)|)} \right| \quad (7)$$

$$FDM_3(f) = \left| \frac{|Hi_1''(f)| - |Hi_2''(f)|}{\frac{7,2}{N} \sum_{i=1}^N (|Hi_1''(i)| + |Hi_2''(i)|)} \right| \quad (8)$$

$$FDM(f) = 2 \cdot |FDM_1(f) + FDM_2(f) + FDM_3(f)| \quad (9)$$

kde $Lo_1(f)$ a $Lo_2(f)$ jsou hodnoty 1. respektive 2. průběhu v bodě f . Lo značí nižší frekvenční pásmo získané zpětnou Fourierovou transformací, jak je popsáno výše. $Hi_1(f)$ a $Hi_2(f)$ jsou hodnoty vyššího frekvenčního pásma v bodě f . Označení s jednou čárkou (např. $Lo_1'(f)$) značí první derivaci, dvě čárky pak znamenají druhou derivaci a N je počet prvků ve frekvenčním pásmu, se kterým zrovna počítáme.

Výsledkem jsou funkce $ADM(f)$ a $FDM(f)$. ADM (z angličtiny Amplitude Difference Measure) znázorňuje míru shody pomalu se měnících částí průběhů v závislosti na frekvenci, FDM (Feature Difference Measure) se naopak zabývá rychle se měnícími partiemi. Čím větší jsou hodnoty těchto funkcí (ADM i FDM), tím větší je neshoda mezi vstupními průběhy. Z těchto funkcí lze ještě vypočítat třetí funkci GDM (Global Difference Measure) podle vztahu [1]:

$$GDM(f) = \sqrt{ADM(f)^2 + FDM(f)^2} \quad (10)$$

GDM zohledňuje data z ADM i FDM a slouží pro celkové zhodnocení.

Metoda FSV je tedy o poznání komplexnější a lépe reprezentuje shodu nebo rozdílnost zkoumaných křivek. Užitečnost jejího použití pro porovnávání křivek s vysokou hustotou hodnot a především průběhů s rychle se měnícími hodnotami je však omezené. Je to způsobeno tím, že metoda využívá numerické derivace. Při srovnávání průběhů s rychle se měnícími hodnotami, kde se prakticky nenachází dva sousední body se stejnou hodnotou, vznikají při použití derivace velké numerické chyby a metoda se tak stává pro tyto případy nepřesnou.

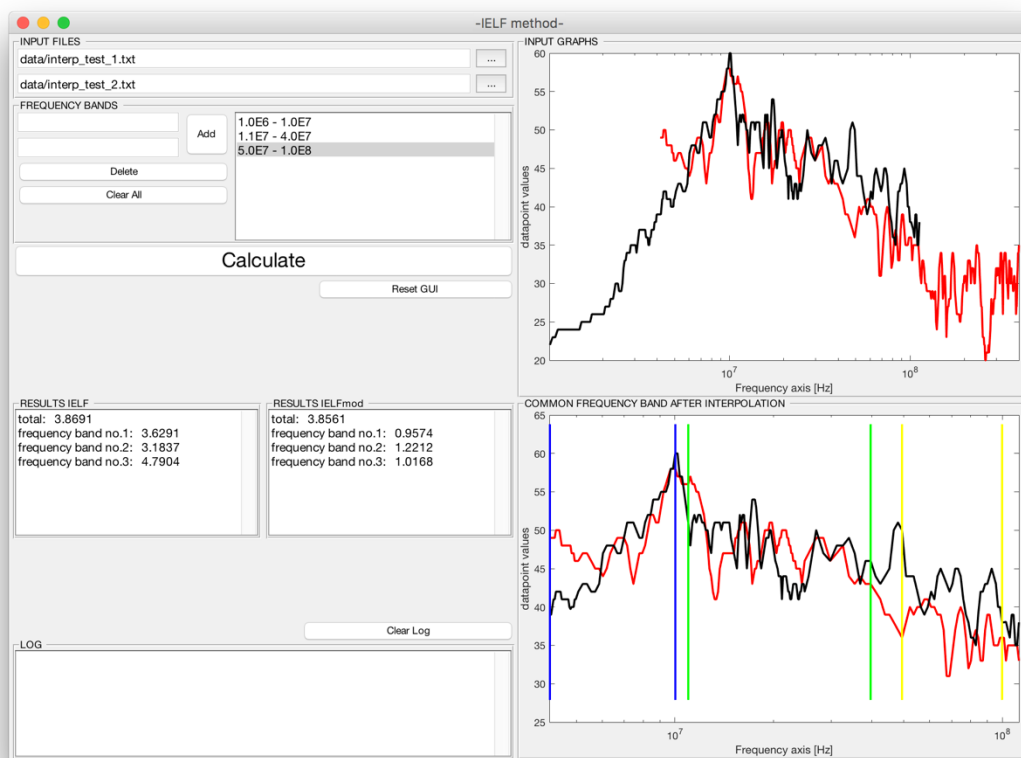
2 GRAFICKÉ UŽIVATELSKÉ ROZHRANÍ

Tato kapitola se věnuje popisu výsledné aplikace z hlediska uživatele. Obsahuje popis funkce, ovládání a možností použití. Aplikace se spouští souborem *Gui.m*.

Velikost okna aplikace je při spuštění vždy upravena podle rozlišení používaného monitoru a okno je možné zvětšovat a zmenšovat. Omezena je pouze minimální velikost okna a to na rozlišení 1000x560 pixelů, což by neměl být ve 21. století problém, neboť monitory s nižším rozlišením se už prakticky nevyskytují.

2.1 Podoba grafického rozhraní

Konečná podoba aplikace je na obrázku 1. V pravé polovině okna se nachází dva grafy zobrazující vstupní data (*INPUT GRAPHS*) a data po interpolaci (*COMMON FREQUENCY BAND AFTER INTERPOLATION*), neboli společnou frekvenční oblast vstupních průběhů, společně s vyznačením okrajů zadáných frekvenčních pásem. Levá horní čtvrtina okna obsahuje panel s řádky pro zadání cest ke vstupním souborům (*INPUT FILES*) a panel s ovládacími prvky pro zadávání požadovaných frekvenčních pásem (*FREQUENCY BANDS*), pro které bude proveden výpočet koeficientů metody IELF. Levá dolní čtvrtina je věnována panelům pro výpis výsledků výpočtů (*RESULTS IELF* a *RESULTS IELFmod*) a panelu *LOG*, kam jsou vypisovány chybové hlášky v případě zadání chybné cesty k souboru apod.



Obrázek 1 Výsledná podoba aplikace

2.2 Vstupní soubory

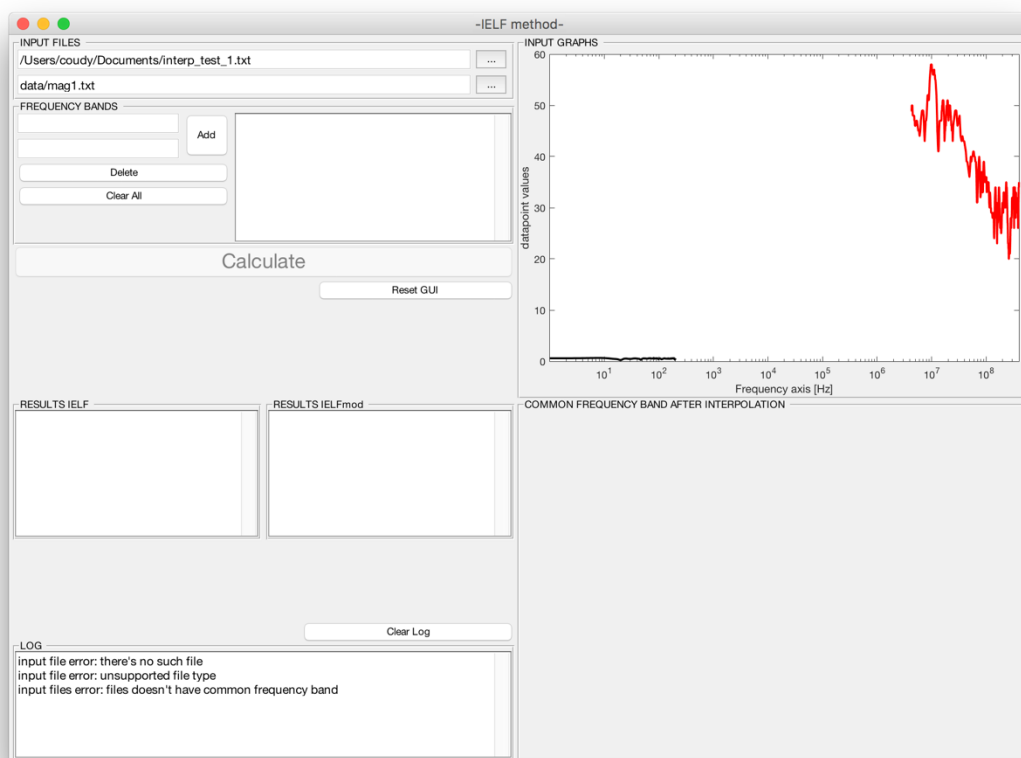
Panel pro zadávání cest ke vstupním souborům je nadepsaný *INPUT FILES* a nachází se v levém horním rohu okna. Obsahuje dva řádky umožňující editaci textu a dvě tlačítka. Cestu k vstupnímu souboru je možné zadat dvěma způsoby:

1. zapsáním cesty k souboru ručně, buď absolutně nebo relativně k spouštěcímu souboru *Gui.m*, jak je vidět na obrázku 2
2. použitím tlačítka na konci řádku (...), které vyvolá klasické dialogové okno pro výběr souboru

Po vložení cesty k souboru jedním z výše uvedených možností dojde k vykreslení průběhu ze zadaného souboru do příslušného grafu. Po zadání druhého souboru je do stejného grafu přidán i druhý průběh. Při smazání jedné nebo obou cest je z grafu průběh (resp. průběhy) odebrán. Aplikace umožňuje načítání souborů následujících datových typů: *txt*, *xls*, *xlsx* a *csv*. Vstupní soubory nemusí být stejného datového typu a nemusí být stejně dlouhé ani stejně vzorkované.

Při zadávání cest k souborům mohou být v panelu *LOG* zobrazeny 3 různé chybové hlášky:

- *input file error: there's no such file:*
tato chybová hláška se objeví v případě, že je cesta zadána chybně a nevede tak k žádnému souboru, nastává většinou v případě zadávání cesty k souboru ručně
- *input file error: unsupported file type:*
tato chybová hláška se objeví při zadání cesty k souboru nepodporovaného datového typu
- *input files error: files doesn't have common frequency band:*
tato chybová hláška se objeví v případě, že zadané průběhy nemají společnou frekvenční oblast. V tomto případě dojde k vykreslení obou průběhů do grafu, takže je tato chybějící společná oblast jasně zřetelná i z grafu vstupních souborů.



Obrázek 2 Zadávání vstupních dat

Jsou-li oba soubory načteny bez problému, tj. jsou zobrazeny oba průběhy a mají společnou frekvenční oblast, dojde k povolení tlačítka *Calculate* a je možné spustit výpočet.

2.3 Frekvenční pásma

Pro zadávání frekvenčních pásem, ve kterých bude proveden výpočet míry shody je v aplikaci určen panel *FREQUENCY BANDS*. Není-li zadáno ani jedno pásmo, dojde po spuštění výpočtu k zobrazení pouze koeficientu pro celý frekvenční rozsah.

Dva řádky pro vkládání textu jsou určeny pro zapsání dolní a horní meze požadovaného frekvenčního pásma. Tlačítkem *Add* pak dojde k zapsání tohoto pásma do tabulky vpravo. Zadáním každého nového pásma se vytvoří nový řádek s novým pásmem, počet frekvenčních pásem není prakticky omezen. Tlačítko *Delete* slouží ke smazání pásma, které je zrovna označeno v tabulce, je možné označit a smazat i více pásem najednou. Je-li potřeba smazat všechna vložená pásma, je možné použít tlačítko *Clear All*, nebo také označit všechna pásma a použít výše zmíněné tlačítko *Delete*.

Hodnoty mezi frekvenčního pásma je možné zadávat zapsáním celého čísla (např. 4 125 000) nebo zkráceně (např. 4.125E6), v obou případech je nutné použít desetinnou tečku, což vychází z amerického původu programu MATLAB. Při zadání desetinné čárky nebo jiné chyby v zápisu frekvenční meze dojde v panelu *LOG* k zobrazení chybové hlášky *freq. band error: limit of frequency band is in incorrect format*.

2.4 Spuštění výpočtu a výpis výsledků

Samotný výpočet se spouští tlačítkem *Calculate*. Toto tlačítko je povolené pouze v případě že jsou úspěšně načteny oba vstupní soubory a mají společnou frekvenční oblast.

Po spuštění výpočtu dojde napřed k vykreslení druhého grafu. Zobrazí se frekvenční oblast společná pro oba průběhy. V případě, že vstupní průběhy nejsou stejně vzorkované nebo jsou frekvenční osy jinak dlouhé, dojde k interpolaci. Při interpolaci je vždy zachována frekvenční osa toho průběhu, který má ve společné oblasti více bodů (je hustěji vzorkována) Tím dojde k minimalizaci numerických chyb při interpolaci. Druhý průběh je pak interpolován, aby se jednotlivé body na frekvenční ose shodovaly. V druhém grafu se zároveň zobrazí barevné svislé čáry reprezentující meze zadaných frekvenčních pásem.

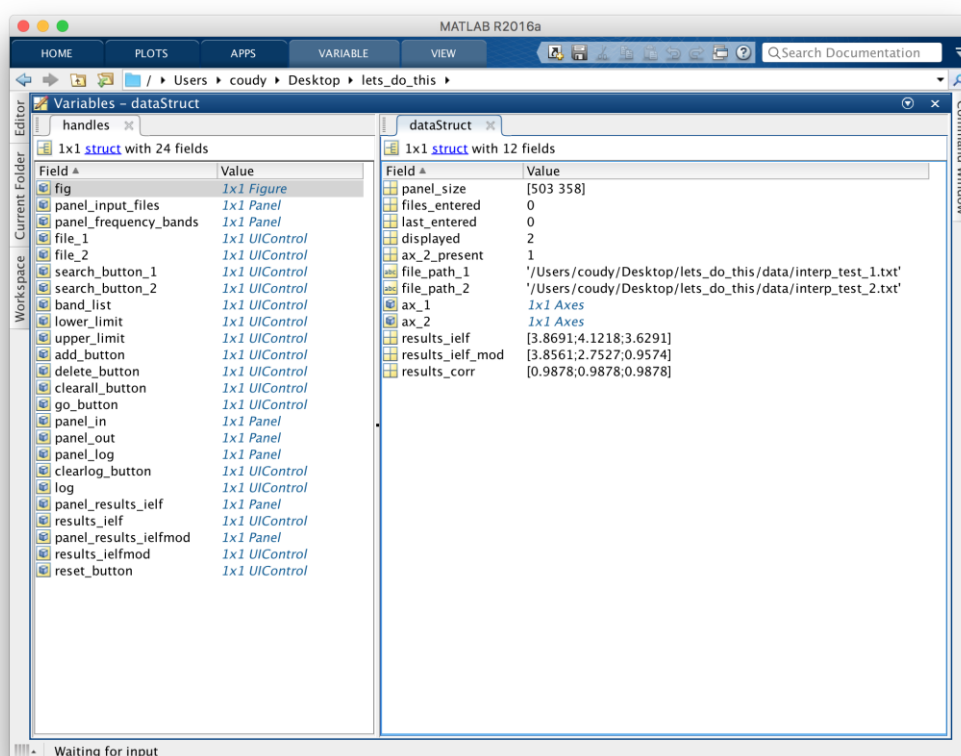
Následně konečně dojde k volání funkce pro výpočet IELF koeficientů. Výsledky provedených výpočtů se zobrazují v panelech *RESULTS IELF* a *RESULTS IELFmod*. V prvním řádku obou polí se vždy zobrazí výsledný koeficient pro celý rozsah společné frekvenční oblasti vstupních průběhů. Tento řádek se zobrazí vždy, bez ohledu na počet zadaných frekvenčních pásem. Následují řádky s koeficienty pro jednotlivá zadaná pásma, které se zobrazí ve stejném pořadí, v jakém jsou frekvenční pásma v tabulce pro jejich zadávání. Počet řádků v tabulce výsledků bude tedy vždy o jedna větší, než je počet zadaných pásem.

3 PROGRAM V POZADÍ

Tato kapitola popisuje program v pozadí grafického rozhraní. Jsou zde popsány jednotlivé funkce zajišťující činnost aplikace a jiné části kódu.

Program byl vytvořen bez použití nástroje *Guide* [6], který je vhodný pro jednodušší aplikace a umožňuje snadné a rychlé definování a rozmísťování ovládacích prvků. Druhou možností je definovat jednotlivé objekty a prvky ručně. To má za výhodu snížení počtu takzvaných *callbacků*, což jsou funkce obsluhující interakce s daným prvkem. Při použití nástroje *Guide* jsou generovány automaticky a ve velkém počtu. Při ručním definování je tedy možné použít pouze funkce, které jsou potřeba, čímž se zjednoduší a zpřehlední výsledný kód.

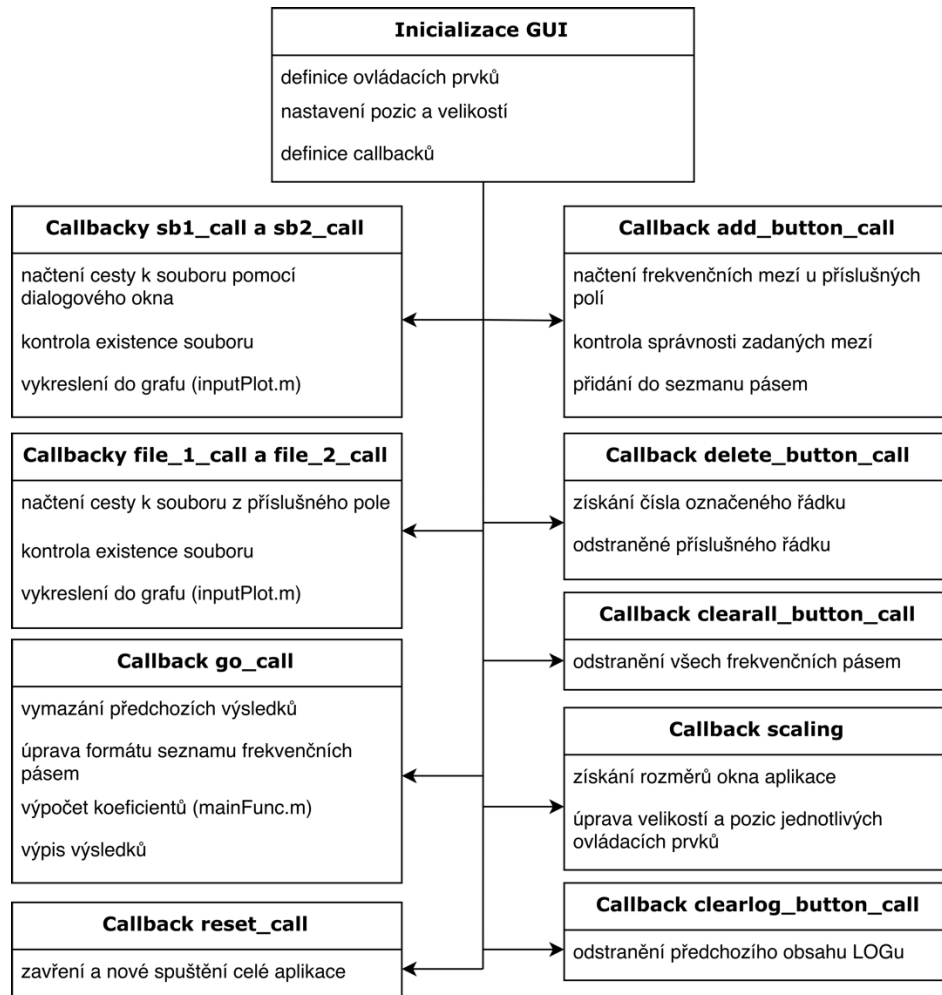
Při tvorbě grafického uživatelského rozhraní v MATLABu jsou pro předávání dat mezi jednotlivými *callbacks* používány datové struktury. Práce se strukturami se však některých verzích Matlabu trochu liší. Proto nebylo prakticky možné použít jedinou strukturu, jak tomu obvykle bývá, ale dvě. Ve výsledném kódu se tedy vyskytují struktury *handles* a *dataStruct*. Struktura *handles* slouží pro uložení dat ovládacích prvků, které vzniknou už při inicializaci aplikace a později se u nich mění pouze některé parametry. Struktura *dataStruct* je naopak určena k ukládání dat, které vzniknou až za běhu aplikace a jsou předávány mezi funkcemi a podobně. Na obrázku 3 jsou pro lepší představu zobrazeny obsahy obou struktur ve chvíli, když už byly provedeny všechny výpočty a vypsány výsledky.



Obrázek 3 Obsah používaných struktur

3.1 Gui.m

Hlavním a nejdelším souborem celé aplikace je *Gui.m*, kterým se spouští grafické rozhraní. Obsahuje definici podoby grafického rozhraní po spuštění aplikace a funkce volané při různých interakcích s ovládacími prvky, tzv. callbacky (viz. obrázek 4).



Obrázek 4 Blokové schéma souboru *Gui.m*

3.1.1 Callbacky *sb1_call* a *sb2_call*

Tyto dva callbacky obsluhují odezvy tlačítek *search_button_1* a *search_button_2*.

Po stisknutí tlačítka dojde nejprve k vyvolání dialogového okna pro výběr souboru ze souborového systému počítače. Cesta ke zvolenému souboru je uložena do proměnné *dataStruct.file_path_1* a následně je otestováno, zda zvolená cesta vede k souboru. Pokud ne, je vypsána chybová hláška *input file error: there's no such file*. Nakonec je volána funkce *inputPlot.m* (viz. kapitola 3.2) pro vykreslení průběhu ze zvoleného souboru do prvního grafu grafického rozhraní.

Callback pro *search_button_2* je téměř totožný, jediný rozdíl spočívá v tom, že pro uložení cesty k souboru používá proměnnou *file_path_2*.

3.1.2 Callbacky *file_1_call* a *file_2_call*

Slouží pro obsluhu dvou řádků pro zapsání cesty k souboru *file_1* a *file_2*. Tyto řádky pro editaci textu mají skoro stejnou funkci jako výše popsaná tlačítka *search_button_1* a *search_button_2*, jen je cesta k souboru získána z uživatelského zápisu místo z funkce pro vyvolání dialogového okna. Funkce pro obsluhu jsou proto velice podobné, dochází zde ke stejné kontrole a volání stejné funkce *inputPlot.m*.

3.1.3 Callback *add_button_call*

Tímto callbackem je ovládána odezva tlačítka *Add* pro přidávání frekvenčních rozsahů. Na začátku jsou načteny hodnoty z řádků pro zadávání mezi nového pásma. Následně je provedena kontrola správnosti získaných dat, testuje se, jestli jsou obě meze vyplněny a jestli jsou zadány numerické hodnoty. V případě, že je jedno nebo obě frekvenční pásma prázdné, nedojde při stisknutí tlačítka *Add* k žádné odezvě, pokud je zadán textový řetězec nebo číslo nesprávného formátu (typicky se jedná a chybu v použití desetinné čárky namísto tečky), je vypsána chybová hláška *freq. band error: entered limit of frequency band is in incorrect format*. Obě načtené meze jsou potom spojeny do jednoho řetězce a přidány do tabulky obsahující seznam frekvenčních pásem. Nakonec je vymazán obsah řádků pro zadávání pásem, aby mohl uživatel pokračovat v zadávání dalšího pásma bez nutnosti manuálně mazat předchozí zadané hodnoty.

3.1.4 Callback *delete_button*

Tlačítko *Delete* slouží ke smazání vybraného frekvenčního rozsahu. Jeho callback je poměrně jednoduchý. Dojde k načtení aktuální podoby seznamu s frekvenčními pásmy a čísla označeného řádku. Následuje podmíněné větvení. Bylo-li dříve zadáno pouze jedno pásmo, je do obsahu tabulky s frekvenčními pásmy zapsán prázdný vektor, čímž dojde k jejímu vyprázdnění. Je-li zadáno více pásem, je na základě dříve načtené informací o označeném řádku smazán pouze tento konkrétní řádek. Získaná upravená verze seznamu frekvenčních pásem je nakonec předána zpět do tabulky pro zobrazení pásem.

3.1.5 Callbacky *clearall_call* a *clearlog_call*

Tyto dva callbacky jsou velice podobné a jednoduché. Slouží ke smazání seznamu frekvenčních pásem, resp. ke smazání obsahu *LOGu*. Dochází zde pouze k vytvoření prázdného textového řetězce a jeho předání k zobrazení v seznamu frekvenčních pásem nebo v *LOGu*.

3.1.6 Callback *reset_call*

Jedná se o další jednoduchou obslužnou funkci. Je provedeno zavření okna s grafickým rozhraním a novému spuštění aplikace. Tím dojde ke smazání všech dříve zadaných dat a uživatel může zadat nové soubory k porovnání apod.

3.1.7 Callback *scaling*

Callback *scaling* je volán při každé změně velikosti okna aplikace. Zajišťuje tedy korektní zobrazení aplikace.

V první řadě je načten parametr samotného okna aplikace, jménem *position*. Z něho lze získat aktuální rozměry okna v pixelech, které jsou následně použity pro změnu velikostí a pozic jednotlivých ovládacích prvků. U většiny z nich dochází pouze ke změně pozice, u grafu pak dochází i ke změně velikosti.

Minimální velikost okna je omezena na 1000x560 pixelů. Při menším okně už začínají být grafy velice malé a nepřehledné.

Kopie výše popsaného callbacku je vložena také funkce inicializace aplikace. I zde plní stejnou funkci, po spuštění je totiž velikost okna nastavena relativně k velikosti zobrazovací jednotky. Jeho velikost je následně převedena na pixely a ze získaných rozměrů jsou dopočítány pozice a velikosti jednotlivých prvků grafického rozhraní.

3.1.8 Callback *go_call*

Nejdelší callback se jmenuje *go_call* a je volán při stisknutí tlačítka *Calculate*, neboli při spuštění aplikace.

Začíná smazáním hodnot výsledků, pro případ že již byla dříve aplikace spuštěna. Dále jsou do lokálních proměnných načteny cesty k souborům a zadané frekvenční pásma (uloženo do proměnné *bands*, viz. kód níže). Pásma nejsou získána jako čísla, ale jako řetězce (datový typ *char* v MATLABu), je proto potřeba provést úpravy pro jejich převedení na číselné hodnoty pro další použití. Je toho docíleno použitím funkce pro rozdělování textových řetězců *strsplit* a funkcí pro převod textu na čísla *str2num* a *cell2num*. Tyto funkce se nachází ve dvou vnořených cyklech, z nichž každý obsahuje řadu podmínek pro ošetření všech možných scénářů. Je to nutné z důvodu podivného chování Matlabu (především při přechodu na jinou verzi programu), například je-li zadáno pouze jedno frekvenční pásmo, jsou data po načtení uložena ve formátu *cell*, je-li frekvenčních pásem více, jsou uloženy jako textové řetězce. Podmínky tedy zajistí, že na konci pole *bands_final* obsahuje dva sloupce číselných hodnot, počet řádků odpovídá počtu zadaných pásem. Celý blok kódu pro úpravu formátu seznamu pásem je následující:

```
for I = 1:band_count
    if (band_count == 1)
        if (iscell(bands))
            b = bands{1,:};
        else
            b = bands;
        end
    else
        b = bands{I,:};
    end
    c = strsplit(b, ' - ');
    for J = 1:2
        e = cell2mat(c(J));
        f = str2num(e);
        bands_final(I,J) = f;
    end
end
```

V dalším kroku je vytvořen druhý graf grafického rozhraní. Zatím nejsou vykresleny průběhy, jsou pouze vytvořeny osy grafu. K zobrazení průběhů dojde později v rámci

funkce *mainFunc.m* (viz. kapitola 3.9). Vykreslování je takto rozděleno z důvodu snazší manipulace s pozicí a velikostí grafu v callbacku *scaling*.

Dále je volána výše zmíněná funkce *mainFunc.m* (viz. kapitola 3.9), do které jsou předány cesty k souborům a konečná verze pole s frekvenčními pásmy. Funkce vrátí pole s vypočítanými koeficienty IELF a IELFmod (*dataStruct.results_ielf* a *dataStruct.results_ielf_mod*).

Poslední částí je výpis výsledků. Data z polí *dataStruct.results_ielf* a *dataStruct.results_ielf_mod* jsou postupně vypsána do odpovídajících seznamů (*RESULTS IELF* a *RESULTS IELFmod*). První je vždy vypsána hodnota koeficientu pro celý frekvenční rozsah a následují koeficienty pro jednotlivá pásma.

3.2 Funkce *inputPlot.m*

Tato funkce zajišťuje vykreslování prvního grafu zobrazujícího průběhy uložené ve vstupních souborech. Má pouze 2 vstupní a jednu výstupní proměnnou, a to struktury *s* a *h* na vstupu a strukturu *s* na výstupu.

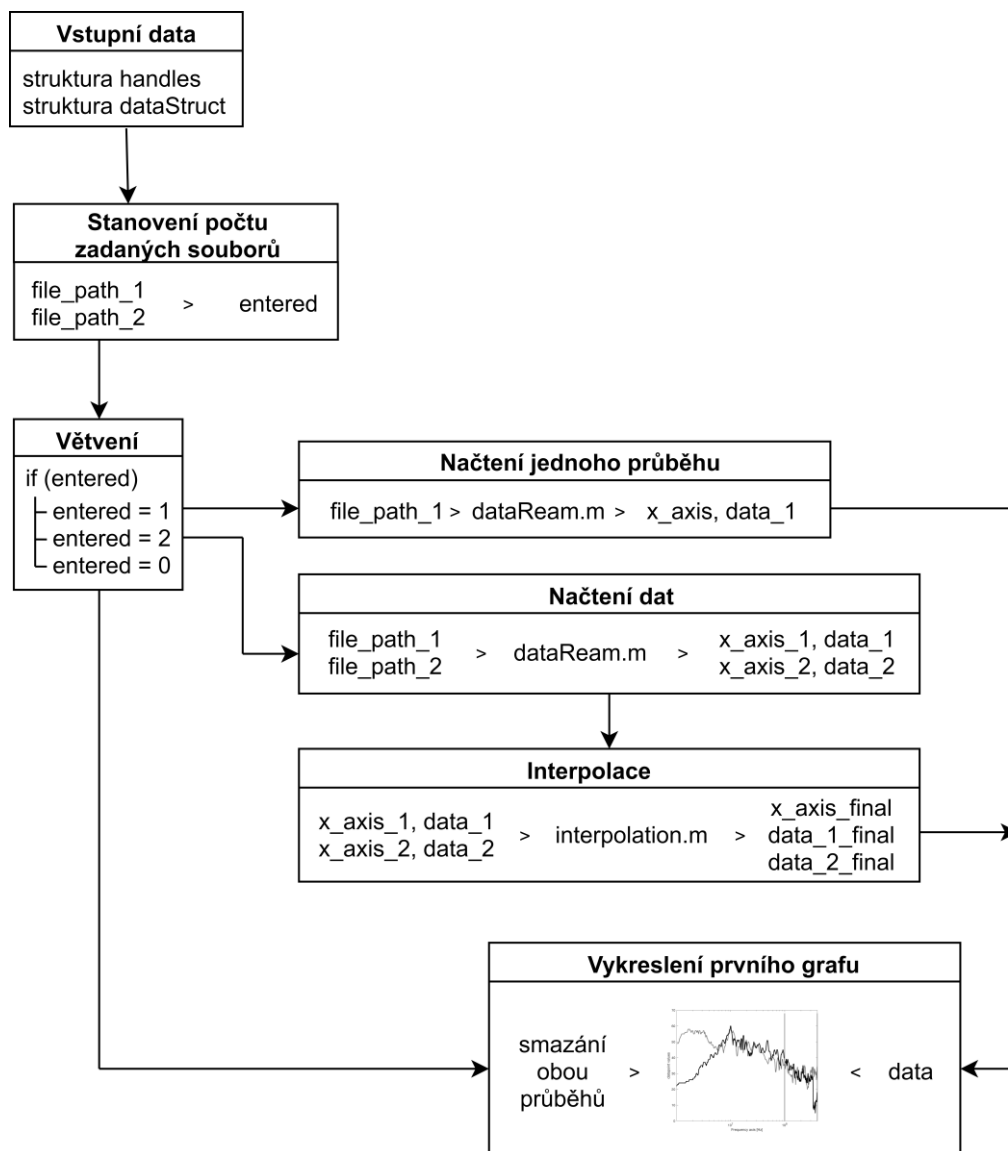
V prvé řadě dochází ke kontrole počtu zadaných vstupních souborů. Těch může být 0 až 2 a kontroluje se obsah proměnných *file_path_1* a *file_path_2*. Pokud je tato proměnná prázdná, znamená to, že vstupní soubor nebyl zadán, pokud ne tak obsahuje cestu k souboru a proměnná *entered* je zvýšena o jedničku. Ukládání nebo mazání cest v těchto proměnných je prováděno callbacky v hlavním souboru *Gui.m*.

Následuje podmíněné větvení programu na základě obsahu proměnné *entered*, neboli počtu zadaných souborů (viz. obrázek 5). Pokud se rovná nule, je zakázáno tlačítko *Calculate* (toto se uplatní v případě, že už byly dříve zadány oba vstupní soubory a potom byl jeden z nich upraven nebo přepsán), funkce *inputPlot.m* je ukončena a nedojde tedy k vykreslení žádného průběhu.

Pokud je *entered* rovno 1 dojde nejprve k načtení dat ze souboru pomocí funkce *dataRead.m* (viz. kapitola 3.3), která může vrátit načtená data nebo chybovou hlášku ohledně nepodporovaného datového typu vstupního souboru. Následuje dotaz na to, zda již byl dříve vykreslen některý průběh, pokud ano dojde k jeho smazání. Dále se provede kontrola výstupu dříve provedené funkce pro načítání dat, pokud vrátila chybovou hlášku, dojde k volání funkce *addToLog.m* (viz kapitola 3.5) se vstupním parametrem obsahujícím chybovou hlášku *input file error: unsupported file type*, tato hláška se vypíše do *LOGu* a funkce *inputPlot.m* je opět ukončena bez vykreslení jediného průběhu. Pokud ale načítání dat vrátilo datové vektory, jsou tato data vykreslena do grafu vstupních průběhů. Na závěr je zakázáno tlačítko *Calculate*, protože je načten pouze jeden datový soubor.

Poslední možností je, že byly načteny dva soubory. V tom to případě dojde napřed ke smazání již existujících zobrazených průběhů, poté konečně k povolení tlačítka *Calculate*. Potom jsou dvakrát volána funkce pro načítání dat (pro načtení dvou vstupních souborů) a je provedena kontrola výstupů funkcí pro načítání dat. Neproběhne-li jedno z načítání dat korektně, je opět vypsána chybová hláška o nepodporovaném formátu dat, správně načtený průběh je vykreslen do grafu a tlačítko *Calculate* je zase zakázáno. V případě, že jsou oba soubory načteny správně, dojde k interpolaci vstupních dat pomocí funkce *interpolation.m* a na základě jejího výstupu může být vypsána chybová

hláška *input files error: files doesn't have common frequency band*. Nakonec jsou do společného grafu vykresleny oba průběhy, čímž funkce *inputPlot.m* končí.



Obrázek 5 Blokové schéma funkce *inputPlot.m*

3.3 Funkce *dataRead.m*

Funkce *dataRead.m* slouží k načítání dat. Vstupní soubory nesmí obsahovat hlavičku a musí být formátovány jednoduše jako dva sloupce dat, první představuje body na frekvenční ose a druhý příslušné hodnoty. Funkce má pouze jeden vstupní parametr, kterým je cesta k souboru který je potřeba načíst a umožňuje načítání dat ze souborů následujících datových typů: *txt*, *xls*, *xlsx* a *csv*. Cesta k souboru je předána ve formě textového řetězce, který je následně rozdělen na menší textové řetězce podle toho, kde se v cestě k souboru nachází tečky. Úsek za poslední tečkou tedy odpovídá datovému typu souboru, ze kterého chceme načítat data. Tento textový řetězec je uložen od proměnné

file_type.

Následují 4 bloky, každý z nich slouží k načítání dat ze souboru jednoho z možných datových typů. Výběr správného bloku je zajištěn přepínačem *switch-case* podle hodnoty uložené v proměnné *file_type*. Pro načítání dat se souborů typu *xls*, *xlsx* a *csv* jsou v MATLABu již předpřipravené funkce, načítání ze souborů typu *txt* je pak řešeno funkcí pro čtení formátovaného textu *fscanf*. Ve všech případech jsou výsledkem dva vektory a jedno pole [1x1]:

1. *x_axis* obsahující hodnoty, které budou později vyneseny na osu *x*,
2. *data* obsahující jednotlivé hodnoty načtené křivky,
3. *len* obsahující délku předchozích dvou vektorů (délky *x_axis* a *data* jsou shodné).

Tyto tři proměnné jsou předány jako výstupní. V případě, že vstupní proměnná obsahuje cestu k souboru jiného datového typu, než jsou 4 výše uvedené, dojde k zobrazení chybové hlášky *input file error: unsupported file type* a všechny výstupní proměnné jsou nastaveny na nulu.

3.4 Funkce *interpolation.m*

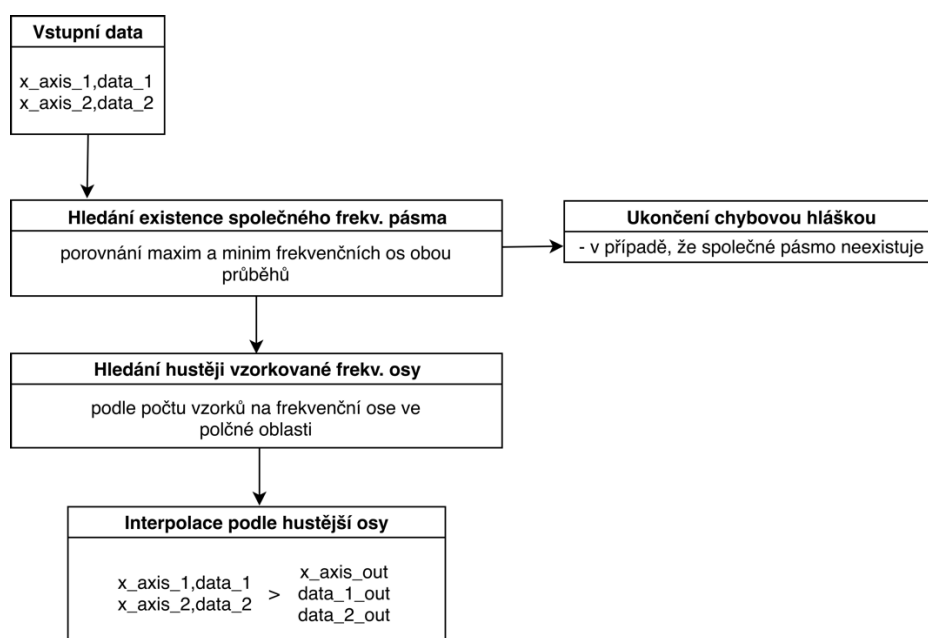
Funkce *interpolation.m* slouží k interpolaci dvou průběhů. Interpolaci je nutné provést v případě, že vstupní průběhy nemají stejnou délku nebo jsou jinak vzorkované. Metoda IELF totiž vyžaduje stejně dlouhé vektory. Funkce má na vstupu dva průběhy, u každého frekvenční osu a vektor s daty, takže celkem 4 vstupní proměnné. Zjednodušené blokové schéma funkce je na obrázku 6.

Napřed je zjištěno, jestli se vstupní frekvenční osy alespoň částečně překrývají, tedy jestli mají společnou frekvenční oblast. Do proměnné *interp_min* je uloženo větší z minim obou průběhů. Do proměnné *interp_max* je naopak uloženo menší z maxim obou průběhů. Zápis pro lepší ilustraci:

```
interp_min = max(min(x_axis_1), min(x_axis_2));  
interp_max = min(max(x_axis_1), max(x_axis_2));
```

V proměnných *interp_min* a *interp_max* jsou tedy uloženy meze společné frekvenční oblasti. Dále dojde k jejich porovnání a pokud platí *interp_min > interp_max*, znamená to, že průběhy nemají společnou frekvenční oblast. Do všech výstupních hodnot jsou v tomto případě uloženy nuly a funkce je ukončena chybovou hláškou *input files error: files doesn't have common frequency band*.

V opačném případě funkce pokračuje hledáním indexů buněk obou vstupních frekvenčních vektorů, které obsahují hodnotu nejbližší mezím společné frekvenční oblasti. Tyto indexy jsou použity pro ořezání vstupních vektorů, vzniknou tedy nové vektory obsahující původní vstupní data, ale pouze ve společné frekvenční oblasti. Porovnáním délek nových vektorů je rozhodnuto, který vstupní průběh bude zachován a který bude interpolován. Zachován zůstane vždy průběh s vyšším počtem hodnot. Samotná interpolace je provedena matlabovskou funkcí *interp1* (jednorozměrná interpolace) a do výstupních proměnných jsou uloženy vektory nové frekvenční osy, dva datové vektory a jeden skalár obsahující délku výstupních vektorů.



Obrázek 6 Blokové schéma funkce *interpolation.m*

3.5 Funkce *addToLog.m*

Jedná se o velice jednoduchou funkci, provádí výpis nového řádku s chybovou hláškou v panelu *LOG*. Má pouze dva vstupní parametry: text který chceme vypsát (*string*) a strukturu *h*, protože jsou prováděny změny v objektu *h.log*.

Nejprve je načten obsah objektu *h.log* před provedením změn. Nebylo-li do *LOGu* ještě nic zapsáno, je vstupní parametr *string* jednoduše překopírován do *h.log*. Pokud již naopak byly zobrazeny nějaké chybové hlášky, dojde k přidání nového řádku k původnímu obsahu a nová výsledná verze je opět předána do *h.log*.

3.6 Funkce *bandCheck.m*

Jak název napovídá, jedná se o funkci provádějící kontrolu jednoho frekvenčního pásma. Ve výsledné aplikaci je volána tolikrát, kolik je zadaných pásem, proto kontroluje pouze jedno pásmo. Vstupními proměnnými jsou zde dolní a horní mez daného pásma a frekvenční osa po interpolaci, tedy společná frekvenční oblast vstupních průběhů.

V první řadě je zkontrolováno, zda se dolní a horní mez shodují nebo ne. Pokud ano, není možné pro dané frekvenční pásmo provést výpočet. Je tedy zobrazena chybová hláška *frequency band no.1 error: band limits are equal, can't be calculated*. Číslo (v tomto příkladu 1) vždy odpovídá pořadí prověřovaného pásma. Výstupní proměnná *passed* je nastavena na nulu do *err* je uloženo číslo 1 (označení nalezené chyby). Funkce je poté ukončena. Pokud jsou frekvenční meze různé, funkce pokračuje.

Další na řadě je srovnání mezí z hlediska velikosti. Je-li zjištěno, že dolní mez je větší než horní mez, dojde jednoduše k jejich prohození, aby mohl výpočet koeficientů IELF proběhnout korektně. Předpokládá se, že prohození horní a dolní meze provedl uživatel omylem, není proto zobrazena žádná chybová hláška.

Následují 4 bloky, z nichž každý kontroluje, zda se meze nachází ve společné frekvenční oblasti vstupních průběhů. V případě, že tomu tak není, je daná mez posunuta na začátek nebo konec společné oblasti. Například je-li dolní mez nastavena na 4MHz a nejmenší hodnota na frekvenční ose je 5MHz, je dolní mez posunuta právě na tuto hodnotu. Obdobně proběhne kontrola pro obě meze.

V případě, že jsou obě meze nad nebo pod frekvenční osou, není možné provést výpočet. Pokud je alespoň jedna z nich ve společné frekvenční oblasti, je možné pokračovat. Toto je poslední kontrola v rámci funkce *bandCheck.m*. Jsou-li tedy na začátku této funkce obě meze nad nebo pod společnou frekvenční oblastí, byly již obě posunuty na začátek nebo konec frekvenční osy. Stačí meze porovnat a v případě že jsou stejné, znamená to, že bylo původně celé pásmo mimo frekvenční osu. Dojde k zobrazení chybové hlášky *frequency band no.1 error: band limits are both above or below common frequency band of input graphs, can't be calculated*, výstupní proměnná *passed* je opět vynulována do *err* je zapsáno číslo 2.

V případě, že celá funkce proběhla bez zjištění chyby, je ukončena nastavením *passed* na jedničku a *err* na nulu.

3.7 Funkce *bandLimits.m*

Funkce *bandLimits.m* zajišťuje výpočet dat pro vykreslení svislých barevných čar, které znázorňují meze frekvenčních pásem.

Pro každé pásmo je potřeba vytvořit dvě čáry (dvě meze) a každá z těchto čar je definována dvěma body, takže funkce má celkem čtyři výstupní proměnné. Těmto proměnným jsou jednoduše přiřazeny odpovídající hodnoty z frekvenční osy. Výška čar je určena podle nejnižší a nejvyšší hodnoty v datových vektorech, vždy vynásobené určitými koeficienty (*coef_min* a *coef_max*) tak, aby konce čar nebyly vizuálně stejně vysoko jako nejvyšší a nejnižší body grafu. Tyto koeficienty jsou určeny podle toho, jestli je maximum či minimum větší nebo menší než 0, jak je vidět na následujícím kódu:

```
if (minimum > 0)
    coef_min = 0.9;
else
    coef_min = 1.1;
end

if (maximum > 0)
    coef_max = 1.1;
else
    coef_max = 0.9;
end
```

3.8 Funkce *ietfCalc.m* a *ietfModCalc.m*

Tato funkce zajišťuje finální výpočet koeficientů metody IELF. Má pouze jeden výstupní parametr, kterým je samotný koeficient. Vstupními parametry jsou pak frekvenční osa, dva datové vektory, koeficienty na frekvenční ose, mezi kterými bude výpočet proveden (neboli meze frekvenčního pásma) a délka frekvenční osy.

Ve funkci se nachází matlabovský zápis vzorce (1) doplněný o dvě podmínky, které jsou použity v případě, že zvolená frekvenční mez je shodná s první nebo poslední hodnotou z datového vektoru. V tomto případě není možné čistě použít vzorec (1), protože by docházelo k chybám v indexování. Datové vektory obsahují N prvků s indexy 1 až N , do vzorce by tedy byly dosazovány prvky s indexy 0 a $N+1$, které neexistují (v MATLABu jsou vektory indexovány od 1 do N), což samozřejmě vede k chybové hlášce a zastavení programu. Tuto chybu je potřeba ošetřit výše zmíněnými podmínkami. V těle obou podmínek se nachází vzorec (1) jednoduše upravený pro počítání krajních bodů vektoru následujícím způsobem:

$$IELF = \frac{\sum_{i=1}^{n-1} |error_i| \cdot (\ln(f_{i+1}) - \ln(f_i))}{\ln(f_n) - \ln(f_0)} \quad (11)$$

pro výpočet s dolní mezí a

$$IELF = \frac{\sum_{i=1}^{n-1} |error_i| \cdot (\ln(f_i) - \ln(f_{i-1}))}{\ln(f_n) - \ln(f_0)} \quad (12)$$

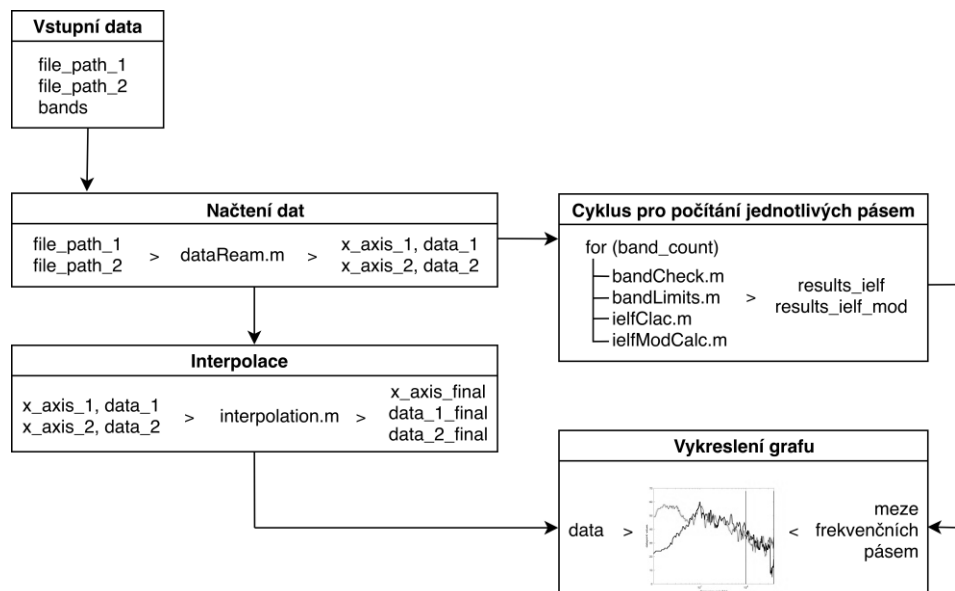
pro výpočet s horní mezí.

Funkce pro výpočet modifikované metody IELF je velice podobná, liší se v použití upraveného vzorce (2). Pro počítání krajních bodů je ve snaze vyhnout se chybám v indexování opět použit lehce modifikovaný původní vzorec, upraven je úplně stejným způsobem jako pro originální metodu IELF.

3.9 Funkce *mainFunc.m*

Funkce vznikla úpravou předchozí verze aplikace, která neobsahovala grafické uživatelské rozhraní, proto není začleněna do souboru s definicí grafického rozhraní. Jedná se o jakousi hlavní funkci, která provádí volání většiny výše popsaných funkcí, vykresluje druhý graf v grafickém rozhraní a zajišťuje počítání IELF koeficientů pro zadaná frekvenční pásma. Výstupními hodnotami jsou zde pouze vektory obsahující výsledky výpočtů pro zadaná pásma a celý frekvenční rozsah. Zjednodušené blokové schéma je na obrázku 7.

Funkce začíná definicí vektoru, obsahujícího názvy barev, které budou později použity pro vykreslení a vizuální odlišení jednotlivých frekvenčních pásem. Dále jsou pomocí funkce *dataRead.m* (viz. kapitola 3.3) načteny oba vstupní soubory. Cesty k nim jsou převzaty ze vstupních proměnných. Následuje interpolace těchto průběhů, je k tomu použita funkce *interpolation.m* (viz. kapitola 3.4). Načítání dat ani interpolace v tuto chvíli nemůže skončit chybovou hláškou, protože už data byla načtena a interpolována při vkládání cest ke vstupním souborům v grafickém rozhraní. Poté dojde k vykreslení společné frekvenční oblasti obou průběhů do druhého grafu grafického rozhraní nadepsaného *COMMON FREQUENCY BAND AFTER INTERPOLATION*.



Obrázek 7 Blokové schéma funkce *mainFunc.m*

Konečně následují výpočty IELF koeficientů, počínaje výpočtem pro celé frekvenční pásmo. Proto jsou do proměnných *lower_index* a *upper_index* uloženy hodnoty odpovídající prvnímu a poslednímu prvku na frekvenční ose. Následně jsou volány funkce *ielCalc.m* (viz. kapitola 3.8) a *ielModCalc.m* (viz. kapitola 3.8), které vrátí příslušné hodnoty koeficientů. Tyto hodnoty jsou uloženy do prvních buněk vektorů *results_ief* a *results_ief_mod*, které budou později vypsány jako výsledky.

Výpočty pro zadaná frekvenční pásma jsou provedeny v rámci velkého cyklu, který proběhne tolikrát, kolik je zadaných pásem. Nejprve jsou z pole *bands* vždy vybrány meze odpovídající pásmu, které je právě počítáno. Tyto meze jsou potom předány jako vstupní parametry funkci *bandCheck.m* (viz. kapitola 3.6) a proběhne kontrola pásma. V případě, že funkce pro kontrolu pásem vrátí chybu, dojde k jejímu zobrazení v panelu *LOG* a do příslušné buňky ve vektorech *results_ief* a *results_ief_mod* je zapsáno NaN. Potom dojde k volání funkce pro získání dat pro vykreslení svislých čas znázorňujících meze pásma *bandLimits.m*. Tyto meze jsou přidány do grafu a cyklus je zakončen předáním příslušných parametrů do funkcí *ielCalc.m* a *ielModCalc.m*, jejichž výstupní parametry (IELF koeficienty) jsou zapsány do vektorů *results_ief* a *results_ief_mod*.

4 OVĚŘENÍ SPRÁVNOSTI

Ověření správnosti výpočtů provedených ve funkci *ielFCalc.m* (viz. kapitola 3.8) bylo provedeno podle již hotového souboru v programu Microsoft Excel. Vzorová data v tomto dokumentu byla získána v rámci projektu HIRF-SE [3].

4.1 Popis vzorového souboru

V souboru jsou již připravené rovnice pro výpočet koeficientu IELF, práce s ním je však velice zdlouhavá a nepřehledná, jak je vidět na obrázku 8 níže.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	freq	A	B		int	abs error	B+delta	int B+				ranges					standard	error vs std	IELF	st
2	1,02E+06	49	22	27	4,07E-01	27	42	1,51E-02	7	1,01E-01	2	112	5,00E+06				55	0	0,00E+00	
3	1,09E+06	49	23	26	7,92E-01	26	43	2,96E-02	6	1,84E-01	3	206	2,00E+07				55	0	0,00E+00	
4	1,09E+06	49	23	26	1,16E+00	26	43	4,37E-02	6	2,65E-01	4	291	1,00E+08				55	0	0,00E+00	
5	1,13E+06	50	24	26	1,52E+00	26	44	5,74E-02	6	3,43E-01	5	486	4,00E+08				55	0	0,00E+00	
6	1,16E+06	52	24	28	1,90E+00	28	44	7,06E-02	8	4,45E-01	6						55	0	0,00E+00	
7	1,20E+06	53	24	29	2,28E+00	29	44	8,34E-02	9	5,56E-01	7	B112	C112				55	0	0,00E+00	
8	1,24E+06	54	24	30	2,66E+00	30	44	9,59E-02	10	6,78E-01	8	B206	C206				55	0	0,00E+00	
9	1,27E+06	55	24	31	3,04E+00	31	44	1,08E-01	11	8,07E-01	9	B291	C291				55	0	0,00E+00	
10	1,31E+06	55	24	31	3,41E+00	31	44	1,20E-01	11	9,34E-01	10	B486	C486				55	0	0,00E+00	
11	1,34E+06	55	24	31	3,77E+00	31	44	1,31E-01	11	1,06E+00	11						55	0	0,00E+00	
12	1,38E+06	55	24	31	4,12E+00	31	44	1,42E-01	11	1,18E+00	12						55	0	0,00E+00	
13	1,42E+06	55	24	31	4,46E+00	31	44	1,53E-01	11	1,29E+00	13						55	0	0,00E+00	
14	1,45E+06	56	24	32	4,81E+00	32	44	1,64E-01	12	1,42E+00	14	averages		delta			55	1	1,06E-02	
15	1,49E+06	56	24	32	5,15E+00	32	44	1,74E-01	12	1,54E+00	15	53	33	20			55	1	2,10E-02	
16	1,52E+06	56	25	31	5,46E+00	31	45	1,85E-01	11	1,65E+00	16	49	50	-1			55	1	3,12E-02	
17	1,56E+06	56	25	31	5,77E+00	31	45	1,94E-01	11	1,75E+00	17	42	43	-1			55	1	4,11E-02	
18	1,60E+06	56	25	31	6,08E+00	31	45	2,04E-01	11	1,86E+00	18	30	26	4			55	1	5,08E-02	
19	1,63E+06	57	25	32	6,38E+00	32	45	2,14E-01	12	1,97E+00	19						55	2	6,97E-02	
20	1,67E+06	58	25	33	6,69E+00	33	45	2,23E-01	13	2,09E+00	20						55	3	9,75E-02	
21	1,70E+06	58	25	33	7,00E+00	33	45	2,32E-01	13	2,20E+00	21						55	3	1,25E-01	
22	1,74E+06	58	26	32	7,28E+00	32	46	2,41E-01	12	2,31E+00	22						55	3	1,51E-01	
23	1,78E+06	58	26	32	7,57E+00	32	46	2,50E-01	12	2,41E+00	23						55	3	1,78E-01	
24	1,81E+06	57	26	31	7,83E+00	31	46	2,58E-01	11	2,50E+00	24						55	2	1,95E-01	
25	1,85E+06	57	26	31	8,10E+00	31	46	2,66E-01	11	2,59E+00	25						55	2	2,11E-01	
26	1,88E+06	56	26	30	8,34E+00	30	46	2,75E-01	10	2,67E+00	26						55	1	2,20E-01	
27	1,92E+06	56	26	30	8,59E+00	30	46	2,83E-01	10	2,75E+00	27						55	1	2,28E-01	
28	1,96E+06	57	26	31	8,84E+00	31	46	2,91E-01	11	2,83E+00	28						55	2	2,44E-01	
29	1,99E+06	57	26	31	9,08E+00	31	46	2,98E-01	11	2,92E+00	29						55	2	2,59E-01	
30	2,03E+06	57	26	31	9,32E+00	31	46	3,06E-01	11	3,00E+00	30						55	2	2,74E-01	
31	2,06E+06	56	27	29	9,54E+00	29	47	3,14E-01	9	3,06E+00	31						55	1	2,82E-01	
32	2,10E+06	55	27	28	9,75E+00	28	47	3,21E-01	8	3,12E+00	32						55	0	2,82E-01	
33	2,14E+06	55	27	28	9,95E+00	28	47	3,28E-01	8	3,18E+00	33						55	0	2,82E-01	
34	2,17E+06	55	27	28	1,02E+01	28	47	3,35E-01	8	3,23E+00	34						55	0	2,82E-01	
35	2,21E+06	56	28	28	1,04E+01	28	48	3,42E-01	8	3,29E+00	35						55	1	2,89E-01	
36	2,24E+06	57	28	29	1,06E+01	29	48	3,49E-01	9	3,35E+00	36						55	2	3,03E-01	
37	2,28E+06	57	28	29	1,08E+01	29	48	3,56E-01	9	3,40E+00	37						55	2	3,16E-01	
38	2,32E+06	57	29	28	1,09E+01	28	49	3,63E-01	8	3,46E+00	38						55	2	3,30E-01	
39	2,35E+06	57	30	27	1,11E+01	27	50	3,69E-01	7	3,50E+00	39						55	2	3,43E-01	
40	2,39E+06	56	30	26	1,13E+01	26	50	3,76E-01	6	3,54E+00	40						55	1	3,49E-01	
41	2,42E+06	56	30	26	1,15E+01	26	50	3,82E-01	6	3,57E+00	41						55	1	3,56E-01	
42	2,46E+06	57	29	28	1,16E+01	28	49	3,89E-01	8	3,62E+00	42						55	2	3,68E-01	
43	2,50E+06	57	29	28	1,18E+01	28	49	3,95E-01	8	3,67E+00	43						55	2	3,81E-01	
44	2,53E+06	56	29	27	1,20E+01	27	49	4,01E-01	7	3,71E+00	44						55	1	3,87E-01	
45	2,57E+06	56	30	26	1,21E+01	26	50	4,07E-01	6	3,75E+00	45						55	1	3,93E-01	
46	2,60E+06	55	32	23	1,23E+01	23	52	4,13E-01	3	3,76E+00	46						55	0	3,93E-01	
47	2,64E+06	57	33	24	1,24E+01	24	53	4,19E-01	4	3,78E+00	47						55	2	4,05E-01	
48	2,68E+06	57	33	24	1,26E+01	24	53	4,25E-01	4	3,81E+00	48						55	2	4,16E-01	

Obrázek 8 Náhled souboru pro porovnání

Dokument obsahuje 10 dvojic datových křivek (v každém listu jeden pár) připravených k porovnání. Výpočet probíhá pro 4 frekvenční pásma a také pro celý rozsah. Frekvenční rozsahy není možné změnit a vkládání vlastních dat je taktéž prakticky nemožné. Každý list obsahuje na konci semilogaritmický graf zobrazujícím průběh křivek pro vizuální kontrolu.

4.2 Testování

Výše zmíněné průběhy byly z Excelu postupně vyexportovány do souborů všech datových typů, které lze do vytvořeného programu v MATLABu vložit. Při exportu dat je potřeba si dávat pozor na počet zobrazených desetinných míst v buňkách, aby nedošlo k zaokrouhlení a vzniku numerických chyb. Takto byla otestována funkčnost načítání dat a současně správnost počítání výstupních koeficientů. Výsledky však byly zprvu nepřesné, odchylka od správných výsledků se pohybovala v řádu několika desetin, což je poměrně velká chyba, kterou nelze zanedbat. Pomocí ladících nástrojů dostupných v Matlabu (především pomocí krokování) byla odhalena nutnost použití vzorců (11) a (12) pro výpočet v případě, že se jedna (nebo obě) z frekvenčních mezí shoduje s krajním bodem datového vektoru. Tato problematika je popsána výše v kapitole 3.8. Po odstranění této chyby se již výsledky z Excelu i Matlabu shodují.

5 ZÁVĚR

Během bakalářské práce bylo docíleno splnění zadání a vytvoření aplikace s grafickým rozhraním pro prostředí MATLAB. Výsledná aplikace je plně funkční a spustitelná ve všech verzích MATLABu bez přídatných toolboxů. Umožňuje provést výpočet míry shody reprezentovaný koeficientem metody IELF pro dva zadané vstupní soubory. Aplikace obsahuje funkci pro interpolaci vstupních dat, je tak odstraněna nutnost zadávání stejně dlouhých a vzorkovaných dat. Výpočet probíhá pro uživatelem zadaná frekvenční pásma, jejichž počet není prakticky omezen, a také pro celý frekvenční rozsah.

LITERATURA

- [1] 2005 IEEE INTERNATIONAL SYMPOSIUM ON ELECTROMAGNETIC COMPATIBILITY, 8-12 August 2005 a [SPONSORED BY] IEEE EMC SOCIETY. *2005 IEEE International Symposium on Electromagnetic Compatibility: EMC 2005*. New York: Institute of Electrical and Electronics Engineers, 2005. ISBN 0780393805.
- [2] IEEE EMC SOCIETY. *2003 IEEE International Symposium on Electromagnetic Compatibility: symposium record : Boston, August 18-22, 2003*. Piscataway, N.J: IEEE, 2003. ISBN 0780378350.
- [3] *HIRF-SE project* [online]. 2017 [cit. 2017-12-11]. Dostupné z: <http://hirfse.axessim.eu>
- [4] Korelace. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2017-12-11]. Dostupné z: <https://cs.wikipedia.org/wiki/Korelace>
- [5] SOCIETY, EMC a IEEE. *2010 IEEE International Symposium on Electromagnetic Compatibility (EMC 2010), Fort Lauderdale, Florida, USA, 25-30 July 2010*. Piscataway, N.Y: IEEE Service Center, 2010. ISBN 9781424463077.
- [6] *MathWorks - Makers of MATLAB and Simulink - MATLAB & Simulink* [online]. Copyright © 1994 [cit. 2018-5-21]. Dostupné z: https://www.mathworks.com/help/matlab/creating_guis/about-the-simple-guide-gui-example.html

SEZNAM ZKRATEK

<i>ADM</i>	Amplitude Difference Measure, funkce pro míru shody pomalu se měnících částí průběhů
<i>FDM</i>	Feature Difference Measure, funkce pro míru shody rychle se měnících částí průběhů
<i>GDM</i>	Global Difference Measure, funkce pro celkové zhodnocení podobnosti průběhů
EMC	Electromagnetic compatibility, elektromagnetická kompatibilita